A Cohesive Finite Element Formulation Using the Finite Cell Method

Nickolaus J. Sundholm
University of Minnesota: Department of Civil, Environmental and Geo Engineering
(Dated: October 1, 2015)

Abstract This report was generated as result of the undergraduate research opportunity program (UROP) at the University of Minnesota. It provides a brief introduction to the finite element method and presents in detail the formulation of cohesive zone models for the use within traditional finite element geometry meshes. The cohesive zone model created is then implemented within the finite cell method to probe it's feasibility to accurately model delamination and fracture of complex geometries typically seen within voxel representations obtained from medical imaging technologies. Several benchmarks tests are then used to verify the accuracy of the model created including an embedded circle subjected to a standard pull test.

I. INTRODUCTION

Initialized by the work of M.J (Jon) Turner of Boeing over the period 1950-1962 [1], the finite element method (FEM) has transformed from a method only available to those with access to high capacity computers to a powerful tool used across numerous engineering disciplines; approximating systems of differential equations of any size. With the increase in computing power, commercial finite element codes have allowed for rapid prototyping of theories and designs in a way that has transformed the engineering world.

More recently, the core of FEM has been generally accepted to be stable and somewhat impervious to change. However various topics remain to frontier FEM; these topics include multiphysics, multiscale models, symbolic and high performance computation, integrated design and manufacturing, advances in information technology, optimization, high performance elements, materials, inverse problems, and treatment of joints and interfaces. This report deals with the last topic.

II. FINITE ELEMENT OVERVIEW

II.1. Variational Formulation

Finite element methods are based on the variational formulation of partial differential equations. To illustrate the variational formulation, a 1D case will be derived from the Minimal Potential Energy principle. Accepting a basic understanding of the mechanics of materials, it can be shown that the *internal energy density* at a point of a linear-elastic material subjected to a one-dimensional state of stress σ and strain ε is:

$$U = \frac{1}{2}\sigma(x)\varepsilon(x) \tag{1}$$

Where σ is connected to the displacement u through Hooke's law, and the strain displacement relation shown here, respectively:

$$\sigma = E\varepsilon \tag{2}$$

$$\varepsilon = u' = \frac{\partial u}{\partial x} \tag{3}$$

The *total internal energy* is then found by integrating over the volume, or distance L in the case of the 1D example, to obtain the following:

$$U = \frac{1}{2}\int_V \sigma\varepsilon dV = \frac{1}{2}\int_0^L F\varepsilon dx = \frac{1}{2}\int_0^L u'EAu'dx \tag{4}$$

To maintain equilibrium, the total internal energy must be equated by the *external work potential* performed by applied mechanical loads on the element displacements. This potential is denoted as W in the ensuing derivation and has contributions from the applied load q(x) and the respective displacement at the point of the applied load u(x). Thus, the external energy can be concisely expressed as:

$$W = \int_0^L qudx \tag{5}$$

The total potential energy can then be defined as:

$$\Pi = U - W \tag{6}$$

Π is a *functional*, called the *Total Potential Energy* or TPE. It depends only on the axial displacement u(x). In Variational Calculus u(x) is called the primary variable of the functional. According to the rules of Variational Calculus, the Euler-Lagrange equation for Π, denoted as Λ for convenience is:

$$\Lambda = \frac{\partial\Pi}{\partial u} - \frac{d}{dx}\frac{\partial\Pi}{u'} = -q - (EAu')' \tag{7}$$

The stationary condition for Π is $\Lambda = 0$, or:

$$EAu'' + q = 0 \tag{8}$$

This is the strong equation of equilibrium in terms of the axial displacement, however it is instead replaced by $\delta\Pi = 0$ within the FEM development.

II.2. Admissible Variations

The concept of *admissible variations* is fundamental in understanding the variationally formulated FEM. A functional can only have one primary variable that is allowed to vary. In the case of the TPE functional, this primary variable is the displacement along x, u(x). If we add a variation of the primary variable, δu(x), the TPE then becomes:

$$\Pi = \Pi[u] \Rightarrow \Pi + \delta\Pi = \Pi[u + \delta u] \tag{9}$$

The function δu(x) and the scalar $\delta\Pi$ are called the *variations* of u(x) and Π, respectively. A displacement variation δu(x) is said to be admissible when both u(x) and u(x) + δu(x) are *kinematically admissible* in the sense of the Principle of Virtual Work (PVW). For a function to be kinematically admissible, it must obey two conditions: it is continuous over the bar length and it satisfies exactly any displacement boundary condition.

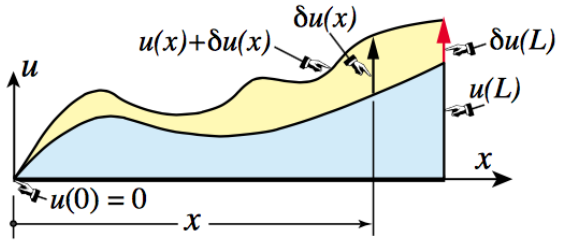

FIG. (1)
Concept of admissible variation of the displacement function u(x)

II.3. The Minimum Total Potential Energy Principle

To effectively estimate a solution, we minimize the total potential energy by stating that the actual displacement solution u*(x) that satisfies the governing equations is that which renders Π stationary:

$$\delta\Pi = \delta U - \delta W = 0 \tag{10}$$

With respect to admissible variations u = u*+δu of the exact displacement field u*(x). This concept is called the Minimum Total Potential Energy Principle (MTPE).

II.4. TPE Discretization and Finite Element Equations

In order to apply the TPE functional to the derivation of FEM equations we must replace the continuous mathematical model with a discrete set of elements that together span the domain of interest. This is a fundamental idea of FEM that allows an accurate solution to be obtained. Figure 2 illustrates the subdivision of a fixed-free bar member discretized into four two-node elements.

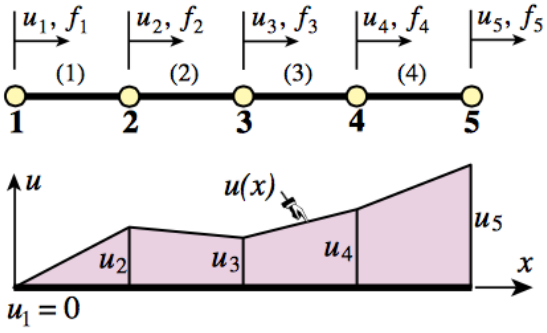

FIG. (2)
FEM discretization of line member

Functionals are scalars, therefore the TPE may be decomposed into a sum of contributions from each individual element:

$$\Pi = \Pi(1) + \Pi(2) + ...\Pi(Ne) \tag{11}$$

In which Ne denotes the number of elements. The same decomposition applies to both its internal energy, external work potential and stationary condition - shown here, respectively:

$$\delta U = \delta U(1) + \delta U(2) + ...\delta U(Ne) = 0 \tag{12}$$

$$\delta W = \delta W(1) + \delta W(2) + ...\delta W(Ne) = 0 \tag{13}$$

$$\delta\Pi = \delta\Pi(1) + \delta\Pi(2) + ...\delta\Pi(Ne) = 0 \tag{14}$$

In linear FEM, the discretization process based on the TPE functional leads to:

$$\Pi(e) = U(e) - W(e) \tag{15}$$

where:

$$U(e) = \frac{1}{2}u(e)^T K(e)u(e) \tag{16}$$

$$W(e) = u(e)^T f(e) \tag{17}$$

K(e) and f(e) are called the element stiffness matrix and the element consistent nodal force vector. Taking the variation of Π(e) with respect to the node displacements gives:

$$\delta\Pi(e) = \delta u(e)^T \frac{\delta\Pi(e)}{\delta u(e)} = \delta u(e)^T [K(e)u(e) - f(e)] = 0 \tag{18}$$

While the variation δu(e) is arbitrary and can take on a non-zero value, the remaining terms must be zero, therefore:

$$K(e)u(e) = f(e) \tag{19}$$

This results in the familiar stiffness equation in the form of Hooke's law.

II.5. Isoparametric Representation

Up to this point, a general introduction to the variational formulation of FEM has been expressed. What is left is to derive an equation for the stiffness of the discretized elements. To do so involves the isoparametric representation of each individual element along with numerical quadrature integration. The combination of these two ideas transformed the field of FEM in the late 1960's. Together they support a good portion of what is presently used in production FEM programs. For simplicity of explanation, and relevance to the preceding cohesive formulation, the isoparametric representation of a 2D bilinear quadrilateral will be investigated (shown in Figure 3). Although, many other elements ranging from 1D to 3D can be similarly represented using this method.

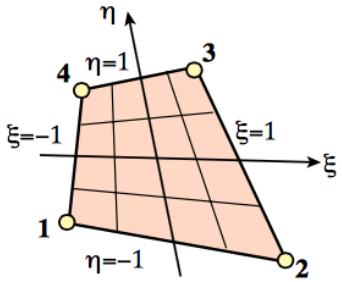

FIG. (3)
The four-node bilinear quadrilateral

The key ides idea is to use shape functions to represent both the element geometry and the problem unknowns. This is done by mapping an element's real coordinates to the natural coordinate system of the element's geometry. The natural coordinates (ξ,η) vary from -1 to +1, taking the value of zero over the quadrilateral medians. This particular variation range was chosen by Irons and coworkers to facilitate use of the standard Gauss integrations formulas. The four-node quadrilateral shown in Figure 3 is the simplest member of the quadrilateral family and is defined by:

$$\begin{bmatrix} 1 \\ x \\ y \\ ux \\ uy \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ x1 & x2 & x3 & x4 \\ y1 & y2 & y3 & y4 \\ ux1 & ux2 & ux3 & ux4 \\ uy1 & uy2 & uy3 & uy4 \end{bmatrix} \begin{bmatrix} N1^e \\ N2^e \\ N3^e \\ N4^e \end{bmatrix} \tag{20}$$

The shape functions vary linearly on quadrilateral coordinates and take on a range of values from 0 to 1 based on the natural coordinates being evaluated. The shape functions are defined for a bilinear quadrilateral here:

$$\begin{bmatrix} N1^e \\ N2^e \\ N3^e \\ N4^e \end{bmatrix} = \begin{bmatrix} \frac{1}{4}(1-\xi)(1-\eta) \\ \frac{1}{4}(1+\xi)(1-\eta) \\ \frac{1}{4}(1+\xi)(1+\eta) \\ \frac{1}{4}(1-\xi)(1+\eta) \end{bmatrix} \tag{21}$$

II.6. The Jacobian and Strain-Displacement Matrix

Partial derivatives of shape function with respect to the Cartesian coordinates x and y are required for the strain and stress calculations. However, because the shape function are not directly functions of x and y but the natural coordinates ξ and η, the quadrilateral element derivations is calculated using the chain rule. Due to conciseness, the derivation of this is skipped and the resulting jacobian is shown here:

$$J = \frac{\partial(x,y)}{\partial(\xi,\eta)} = \begin{bmatrix} \frac{\partial x}{\partial \xi} & \frac{\partial y}{\partial \xi} \\ \frac{\partial x}{\partial \eta} & \frac{\partial y}{\partial \eta} \end{bmatrix} = \begin{bmatrix} \frac{\partial N1^e}{\partial \xi} & \frac{\partial N2^e}{\partial \xi} & \dots & \frac{\partial Nn^e}{\partial \xi} \\ \frac{\partial N1^e}{\partial \eta} & \frac{\partial N2^e}{\partial \eta} & \dots & \frac{\partial Nn^e}{\partial \eta} \end{bmatrix} \begin{bmatrix} x1 & y1 \\ x2 & y2 \\ \dots & \dots \\ xn & yn \end{bmatrix} \tag{22}$$

Somewhat similar to the jacobian, the strain-displacement matrix B computes the strain of an element given the displacements. The full expression is obtained by differentiating the finite element displacement field:

$$\varepsilon = \begin{bmatrix} exx \\ eyy \\ 2exy \end{bmatrix} = Bu^e \tag{23}$$

$$B = \begin{bmatrix} \frac{\partial N1^e}{\partial x} & 0 & \frac{\partial N2^e}{\partial x} & 0 & \dots & \frac{\partial Nn^e}{\partial x} & 0 \\ 0 & \frac{\partial N1^e}{\partial y} & 0 & \frac{\partial N2^e}{\partial y} & \dots & 0 & \frac{\partial Nn^e}{\partial y} \\ \frac{\partial N1^e}{\partial y} & \frac{\partial N1^e}{\partial x} & \frac{\partial N2^e}{\partial y} & \frac{\partial N2^e}{\partial x} & \dots & \frac{\partial Nn^e}{\partial y} & \frac{\partial Nn^e}{\partial x} \end{bmatrix} \tag{24}$$

Noting that u' in equation 4 is the same as the strain ε, the total internal energy of an element can be written as:

$$\Pi^e = \frac{1}{2} u^{eT} K^e u^e - u^{eT} f^e \tag{25}$$

Where the stiffness, K^e, is found by integrating over the volume domain Ω^e (note that the elasticity matrix E is dependent on the properties of the continuum):

$$K^e = \int_{\Omega^e} hB^T EBd\Omega^e \tag{26}$$

And the consistent element nodal force vector is:

$$f^e = \int_{\Omega^e} hN^T bd\Omega^e + \int_{\Gamma^e} N^T td\Gamma^e \tag{27}$$

Where b is the body force acting on the element's volume Ω^e (gravity, electromagnetic potential etc.), t is the traction on the element's boundary Γ^e and N is the shape functions evaluated at the element's natural coordinates. Evaluation of the above integrals is typically done by use of Gauss integration rules. While this idea is not trivial, it is essential to the finite cell method and will be explained in detail in the following sections.

III. THE COHESIVE ELEMENT

The focus of this report will now deflect towards the formulation of cohesive element interface techniques for FEM - which was the main focus of the UROP.

Because the domain over which a solution is to be obtained must first be discretized into finite elements, the accuracy of the solution relies directly on the quality of this discretization. In 2D and 3D problems, this often involves a meshing algorithm used to approximate the actual geometry with a set of finite elements. If the domain consists of various complex geometries, discretizing becomes costly and sometimes results in poor mesh quality, or elements that result in mathematical discontinuities giving infinite stress values - which is inevitably the case for fracture problems where an element interface is split or cracked and a discontinuity arises at the crack nucleation point. Cohesive interfaces can mitigate this discontinuity by defining some prescribed stress distribution based on the crack opening along the fracture plane (see Figure 4).

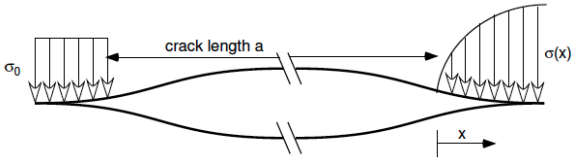

FIG. (4)
Linear and non-linear cohesive traction applied at crack tip

III.1. Traction Separation Law (TSL)

The cohesive model itself is a so called phenomenological model. Therefore, a traction separation law has to be assumed independent from the bulk material as a model quality. An exponential law proposed by Needleman in 1990 is qualitatively shown by Figure 5, and quantitatively by equations 28 and 29.

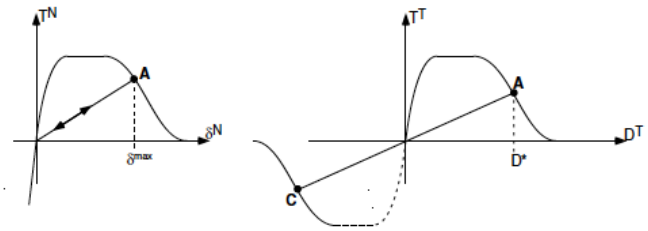

FIG. (5)
Normal and tangential exponential TSL (clevis loading/unloading)

Using this model, tangential and normal separations are handled separately, although they are interconnected by introducing a damage parameter D.

$$Tn = -\frac{27}{4} T^N_{max} (\frac{un}{\delta n})(1-D)^2 \tag{28}$$

$$Tt = -\frac{27}{4} T^T_{max} (\frac{ut}{\delta t})(1-D)^2 \tag{29}$$

$$D = \sqrt{(\frac{un}{\delta n})^2 + (\frac{ut}{\delta t})^2} \tag{30}$$

Where Tn and Tt are the normal and tangential tractions and un, ut, δn, δt, T^N_{max} and T^T_{max} are the normal separation, tangential separation, critical normal separation, critical tangential separation, maximum normal traction and maximum tangential traction, respectively. These values must be predefined based on the cohesive properties of the material to be tested. It is also worth noting that while both the normal and tangential tractions are handled using the same function, the tangential tractions are able to acheive a negative value without penalty. For the normal tractions, however, a negative value means that the interface is in compression. Special care must be taken in this case to allow the traction to go negative, but only follow a straight line as shown by Figure 5, this allows a negative normal traction to "push" the elements apart rather than allowing them to overlap.

III.2. Cohesive Finite Element Formulation

The cohesive element is implemented by inserting "zeros-thickness" elements along a predetermined path. As such, cohesive models are well suited for problems where the fracture path is known ahead of time, such as delamination of composites, and ill-suited for problems where an unknown fracture path is developed.

For 2D geometries, this means that the cohesive element is a 1D line element that passes stiffness to the directly connected 2D elements - thus providing an interface at which a specified amount of stress can be developed.

The additional stiffness of the cohesive elements is accounted for by adding a cohesive term to the total potential energy:

$$\delta\Pi = \delta U - \delta W - \delta\Pi^{coh} \tag{31}$$

Where:

$$\delta\Pi^{coh} = \int_{\Gamma^{coh}} \delta[[u]]t^c d\Gamma^{coh} = \int_{\Gamma^{coh}} [N(u^+ - u^-)]t^c d\Gamma^{coh} \tag{32}$$

From equation 32 we can see that the energy of the cohesive element is gathered from the jump in displacements of the two domains along the cohesive interface. From the assumed TSL, the traction rates can be obtained as:

$$\begin{bmatrix} \dot{T}^t \\ \dot{T}^n \end{bmatrix} = \begin{bmatrix} \partial f1/\partial ut & \partial f1/\partial un \\ \partial f2/\partial ut & \partial f2/\partial un \end{bmatrix} \begin{bmatrix} \dot{ut} \\ \dot{un} \end{bmatrix} = [T] \begin{bmatrix} \dot{ut} \\ \dot{un} \end{bmatrix} \tag{33}$$

Where f1 and f2 are the tangential and normal TSL functions, respectively. If no interaction between the two directions is to be desired, the off-diagonals terms of [T] can be dropped out.

The tangent and normal components of the relative displacement rates can be determined directly from the unit normal vector of the cohesive element. For 2D, the cohesive element can be simplified to a 1D line, as shown in Figure 6.

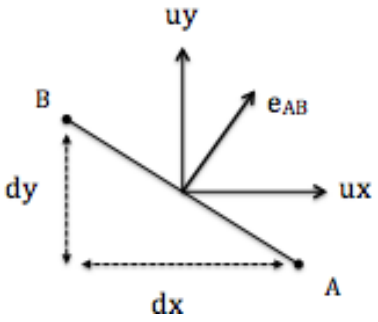

FIG. (6)
Configuration of cohesive interface

The unit normal vector of the line segment $\vec{AB}$ is defined by equation 34. Where l is the length of $\vec{AB}$.

$$\vec{e}_{AB} = \begin{bmatrix} dx/l & dy/l \end{bmatrix} \tag{34}$$

The tangent and normal components of the relative displacement rates between domain 1 and 2 are then related to the unit normal vector by:

$$\begin{bmatrix} \Delta\dot{ut} \\ \Delta\dot{un} \end{bmatrix} = \begin{bmatrix} -\vec{e}_{AB}(1) & \vec{e}_{AB}(2) \\ \vec{e}_{AB}(2) & -\vec{e}_{AB}(1) \end{bmatrix} \begin{bmatrix} \Delta\dot{ux}^{12} \\ \Delta\dot{uy}^{12} \end{bmatrix} = [Q] \begin{bmatrix} \Delta\dot{ux}^{12} \\ \Delta\dot{uy}^{12} \end{bmatrix} \tag{35}$$

Because the cohesive element length is usually small, the unit vector at each Gauss point is approximated by the unit vector of the element. This allows computation at each element to be expedited.

The cartesian components of the relative displacements are then defined by:

$$\begin{bmatrix} \Delta\dot{ux}^{12} \\ \Delta\dot{uy}^{12} \end{bmatrix} = \begin{bmatrix} 1 & 0 & -1 & 0 \\ 0 & 1 & 0 & -1 \end{bmatrix} \begin{bmatrix} \dot{ux}^1 \\ \dot{uy}^1 \\ \dot{ux}^2 \\ \dot{uy}^2 \end{bmatrix} = [A] \begin{bmatrix} \dot{ux}^1 \\ \dot{uy}^1 \\ \dot{ux}^2 \\ \dot{uy}^2 \end{bmatrix} \tag{36}$$

Lastly, the element displacements are mapped to the natural coordinates by the shape function matrix [N]:

$$\begin{bmatrix} \dot{ux}^1 \\ \dot{uy}^1 \\ \dot{ux}^2 \\ \dot{uy}^2 \end{bmatrix} = [N] \begin{bmatrix} \dot{ux}^{1A} \\ \dot{uy}^{1A} \\ \dot{ux}^{1B} \\ \dot{uy}^{1B} \\ \dot{ux}^{2A} \\ \dot{uy}^{2A} \\ \dot{ux}^{2B} \\ \dot{uy}^{2B} \end{bmatrix} \tag{37}$$

Where:

$$[N] = \begin{bmatrix} N1 & 0 & N2 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & N1 & 0 & N2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & N1 & 0 & N2 & 0 \\ 0 & 0 & 0 & 0 & 0 & N1 & 0 & N2 \end{bmatrix} \tag{38}$$

Pulling matrix 33 - 38 together, the elemental stiffness matrix for the cohesive element can be expressed as:

$$[K]^e = \int_{\Gamma^{coh}} [N]^T [A]^T [Q]^T [T] [Q] [A] [N] \, d\Gamma^{coh} \tag{39}$$

In a similar manner, the internal force vector due to cohesive element traction can be derived as:

$$[fcoh]^e = \int_{\Gamma^{coh}} [N]^T [A]^T \begin{bmatrix} Tt \\ Tn \end{bmatrix} \tag{40}$$

Where Tt and Tn are the tangent and normal tractions along the interface. It should be noted that the above integrands are evaluated based on the undeformed configuration, therefore only the matrix [T], Tt and Tn change with loading.

While the cohesive interface element approach for modeling fracture is beneficial in a large number of applications, it requires creating a mesh with zero-thickness elements along the cohesive interface; which is costly and

sometimes error prone. This becomes an increasingly evident problem when the geometry that is to be analyzed is very complex. Because of this, the finite cell method is employed in an effort to extend the above TSL to more complex and arbitrary geometries that do not require any mesh creation or manipulation.

IV. THE FINITE CELL METHOD

The *finite cell method* (FCM), introduced by Parvizian, Duster and Rank in 2007 [2] is an emerging FEM method that is well suited for complex geometries outside the bounds of traditional FEM. It is an embedded domain method, which combines the fictitious domain approach with adaptive integration and weak enforcement of unfitted essential boundary conditions. At its heart, the finite cell method uses a simple structured grid to approximate an embedded geometry. The structured grid is then adaptively refined around the physical domain, allowing Gauss points to be concentrated around the interface between the two domains. Although a detailed introduction of the finite cell method goes beyond the scope of this report, two fundamental concepts will be introduced briefly, namely, the fictitious domain approach and adaptive quadrature.

IV.1. The Fictitious Domain Approach

The fictitious domain approach is achieved by immersing the physical domain within a fictitious domain that closely fits the geometric boundary of the physical domain (see Figure 7). The degrees of freedom of the entire system is then doubled to effectively couple the two domains. Typically, the fictitious domain is to be neglected, or given zero stiffness as the it is outside of the physical domain. To accomplish this interaction with accuracy, a penalization scheme is applied to the fictitious cells that fills the rows and columns of the stiffness equations with zeros. However, in many cohesive interface applications, both the physical and fictitious domains represent real material. Therefore a penalization scheme need not be applied.

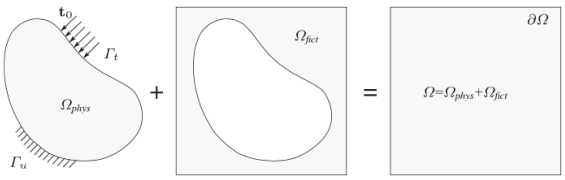

FIG. (7)
The fictitious domain approach: the physical domain Ω^{phys} is extended by the fictitious domain Ω^{fict} into an embedding domain Ω to allow easy meshing of complex geometries

IV.2. Adaptive Quadrature

To obtain a fully coupled system, each Gauss point within the structured mesh must be attributed to the correct domain. When a boundary intersects an element, the accuracy and smoothness of the integrands are directly influenced by the discontinuities that arise from the element attaining stiffness from both domains. To overcome this, the finite cell method uses adaptive Gauss quadrature to improve the integration accuracy in the cells cut by geometric boundaries, based on a hierarchical decomposition of the original cell into integration sub-cells. The refinement process is repeated until a desired refinement depth (k) is attained. At each refinement depth, the Gauss point weights of each sub-cell must hold a value of $1/(2^k)$ to maintain consistent with the iso parameterization of the structured cells. Figure 8 shows a general refinement of a typical embedded geometry.

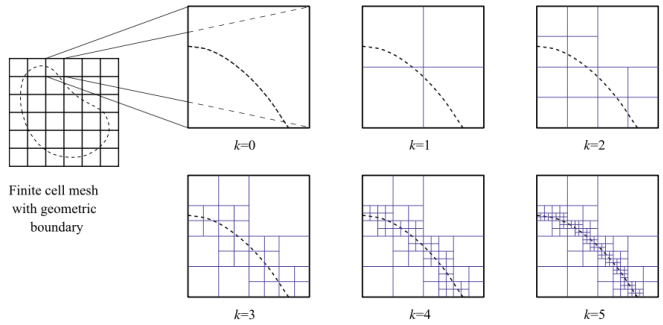

FIG. (8)
2D sub-cell structure for adaptive quadrature integration of finite cells

IV.3. Cohesive Interface Formulation

What is needed now is a way to apply a cohesive TSL at the interface of the embedded geometry. Because the solution field intrinsically holds the information from both geometries, there is no need for specifying which element is cohesive and which is not. For homogeneous materials, simply all that is needed is a surface description of the cohesive interface. The TSL is then implemented at the integration points along the surface, calculating the jump in displacements directly from the two domains.

All elements within the structured grid are quadrilateral elements, and can be either linear or higher order. In the linear case, each element has a total of 16 degrees of freedom (8 for each domain). While the preprocessing of the finite cell domain is different (and more simple) than the traditional cohesive model, the implementation of the TSL, and therefore the cohesive stiffness and constant force vector, is identical to equations 39 and 40 except for matrix [N] - which now becomes a 4x16 matrix:

$$[N]^T_{FCM} = \begin{bmatrix} N1 & 0 & 0 & 0 \\ 0 & N1 & 0 & 0 \\ 0 & 0 & N1 & 0 \\ 0 & 0 & 0 & N1 \\ N2 & 0 & 0 & 0 \\ 0 & N2 & 0 & 0 \\ 0 & 0 & N2 & 0 \\ 0 & 0 & 0 & N2 \\ N3 & 0 & 0 & 0 \\ 0 & N3 & 0 & 0 \\ 0 & 0 & N3 & 0 \\ 0 & 0 & 0 & N3 \\ N4 & 0 & 0 & 0 \\ 0 & N3 & 0 & 0 \\ 0 & 0 & N4 & 0 \\ 0 & 0 & 0 & N4 \end{bmatrix} \quad (41)$$

The stiffness of the FCM cohesive elements and consistent force vector can then be attained by inserting 41 into 39 and 40.

V. TSL VERIFICATION

V.1. Simple Pull Test

To verify that the TSL generated gives expected results, a simple three element pull test is presented as shown in Figure 9. The top and bottom elements are linear quadrilateral finite elements, while the middle node connects the two as a cohesive element. Node N1 and N2 are fixed in the x and y, while N5 and N6 are continuously displaced an incremental amount until the cohesive element releases all traction transfer between the two. The critical normal separation (δ) and the critical normal traction (Tn^{crit}) was chosen to be 0.01 and 10, respectively. The normal traction is plotted versus normal displacement to ensure the solution has converged upon the correct TSL chosen.

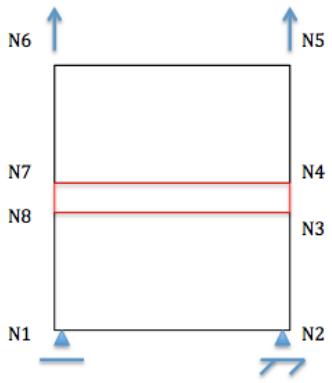

FIG. (9)
Cohesive pull test

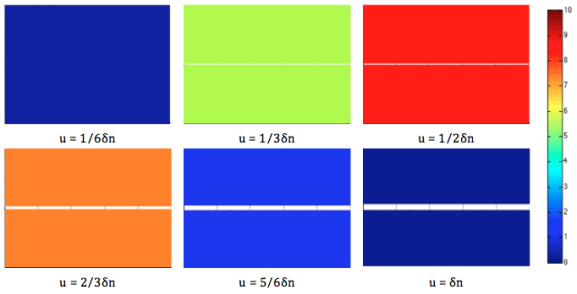

FIG. (10)
Crack growth and stress transferred between elements

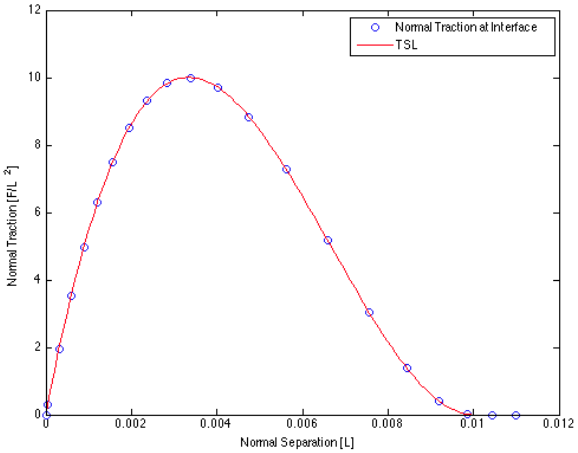

FIG. (11)
Normal traction vs. normal separation

V.2. Loading/Unloading

An area of great concern within many fracture scenarios is the response of a system during cyclic loading (e.g. the growth of a tooth crack during repeated tooth-tooth contact). Because of this, a consistent loading/unloading relationship must be attained within the TSL. Although there are many models that track the maximum damage attained, the benchmark shown here assumes the cohesive interface elastically deforms back to its original damage state during unloading. To show this, the same test is carried out as the simple pull test, except the displacement increments effectively "unload" the elements as shown in Figure 12.

From Figure 13 we see that the traction at the cohesive interface drops off linearly towards zero during unloading. Because the relative or "total" damage is not stored, this model will allow the interface to be loaded and unloading infinitely without failing due to fatigue. This is an extreme idealization that can be changed with a more complex TSL.

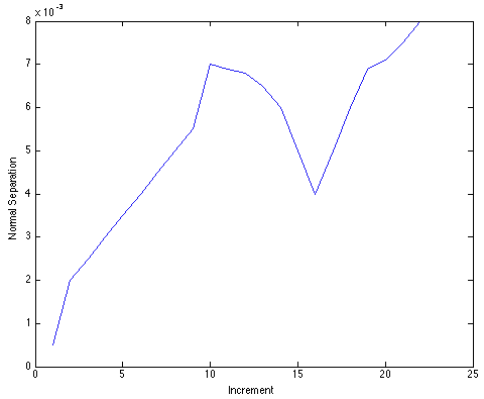

FIG. (12)
Normal separation boundary condition imposed on N5 and N6 for unloading condition

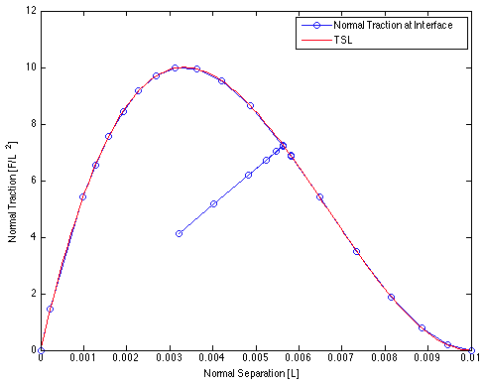

FIG. (13)
Normal separation vs. normal traction during loading/unloading

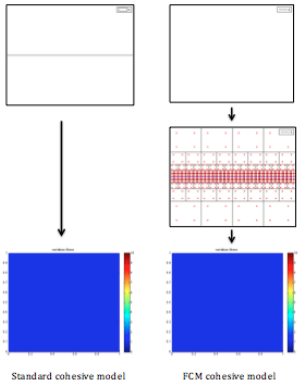

FIG. (14)
Simple pull test to verify FCM cohesive interface (red dots are refined Gauss points near cohesive interface)

VI. FINITE CELL COHESIVE INTERFACE VERIFICATION

To verify that the cohesive interface within the finite cell method gives the correct results, the same pull test shown in Figure 9 is used to compare the standard cohesive model with the cohesive interface using the finite cell method. From Figure 14 it can be seen that the FCM cohesive model produces the exact same results as the standard cohesive model.

VI.1. Multi-Element Immersed Circle

An embedded circle test is computed (Figure 15) to present the advantage of the finite cell method in scenarios with complex geometries. The same loading scheme is used as in Figure 9. In the standard cohesive model, an embedded circle would require the addition of zero-thickness interface elements along the perimeter of the embedded circle. The inside of the circle would also need to be meshed, which is very difficult to do with linear quad elements. The finite cell method allows this geometry to be rapidly created and solved without any pre-computing or segmentation of the original mesh.

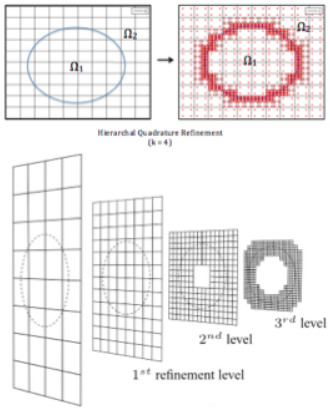

FIG. (15)
Immersed circle example with hierarchical quadrature refinement (Ω_1 and Ω_2 are the same material, with the interface of the two being controlled by the cohesive TSL)

In this particular case, the normal and tangential vector at each cohesive element must be computed to correctly transform the normal and tangential separations from the global solution field (shown visually by Figure 16). This procedure allows for interpretation of complex geometries that include multiple domains. What remains to be seen, however, is it's ability to automatically adapt to various mesh types. At the moment, sufficient thought must be put into the formulation of equations 35 through 38 to correctly calculate the jump between domains. Also, the numbering of the cohesive interfaces must maintain a congruent orientation for the normals to not flip directions. Fortunately, the orientation can be easily verified by constructing Figure 16.

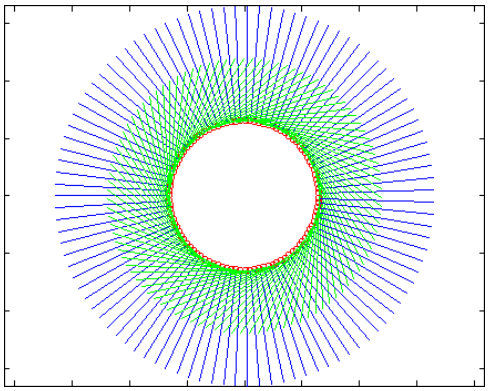

FIG. (16)
Normal (blue) and tangential (green) unit vectors along cohesive line elements

The stiffness of the cohesive elements are calculated at each line element along the circle. For the case where an element is not intersected by the cohesive boundary, the stiffness terms are only attributed to the degrees of freedom for either domain 1 or domain 2. If left unattended, this causes a disconnect within the stiffness matrix that results in an unsolvable system. To mediate this, a 1 is placed on the diagonal everywhere that there isn't a non-zero value (see Figure 17). This effectively connects all the equations in the system without changing the solution vector - essentially mapping the identity matrix on top of the stiffness, resulting in no change.

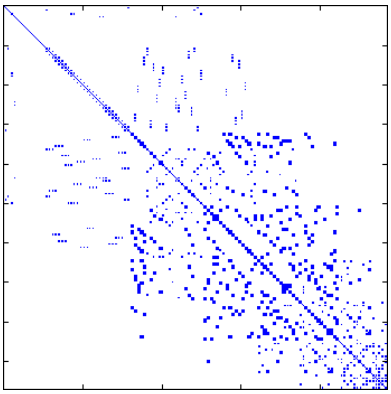

FIG. (17)
Stiffness matrix for multi-element immersed circle (blue indicates non-zero cell value)

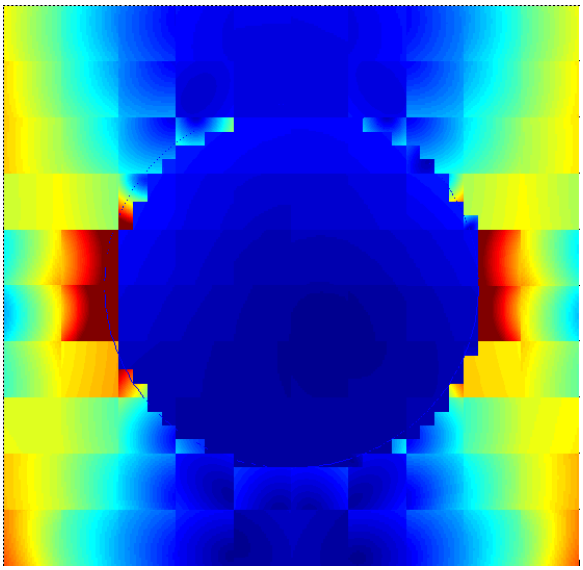

FIG. (18)
von Mises stress of multi-element immersed circle

The stress plot of Figure 18 shows the good and bad of the results obtained. The good is that an expected stress profile has been achieved with a fairly course mesh, which is inevitably connected to the bad. Because the finite cell method refines a structured mesh around the physical boundary, the contour between the the two domains can not be smoothed without visualizing the solution field on a separate mesh that is independent of the original structured mesh. While this isn't a large problem, it adds an added complexity to various problems where a clean mesh is hard to obtain.

VII. CONCLUSION

The preceding report presents a cohesive interface finite element formulation using the finite cell method. It was shown that a cohesive interface feature of the finite cell method holds extreme promise in mitigating the current implementation problems of traditional cohesive element formulations with respect to complex geometries. Because of its ability to handle geometries taken from image based techniques such as medical CT scans, a possible application would be the analysis of a hip socket replacement or the fatigue life of a dental tooth cap. Future work would include a 3D implementation with higher order quad and tetrahedral elements. It would also be interesting to incorporate some sort of ray-casting algorithm within the finite cell method to allow automatic domain detection independent of the geometry chosen. This would open the door to implementing cohesive interfaces between further complex geometries, and ones that contain multiple mediums.

[1] M. J. Turner, R.W. Clough, H. C. Martin and L. J. Topp. *Stiffness and Deflection Analysis of Complex Structures*. Journal of the Aernoautical Sciences, 1956.

[2] J. Parvizian, A. Duster and E. Rank. *Finite cell method: h- and p- extension for embedded domain problems in solid mechanics*. Computational Mechanics, 2007

[3] S Roy Chowdhury and R Narasimhan. *A cohesive finite element formulation for modeling fracture and delamination in solids*. Sadhama, Vol. 25, Part 6, 2000.

[4] C. Felippa. *Introduction to finite element methods*. Department of Aerospace Engineering Sciences, University of Colorado at Boulder.

[5] Ingo Scheider. *Cohesive model for crack propagation analyses of structures with elastic-plastic material behavior*. [*Foundations and implementation*]. GKSS research center Geesthacht, Dept. WMS, 2001.

[6] Nguyen Vinh Phu. *Crack modelling using zero-thickness interface elements*.

[7] N.Zander, T. Bog, M. Elhaddad, R. Espinoza, M. Hu, A. Joly, C. Wu, P. Zerbe, A. Duster, S. Hollmannsberger, J. Parvizian, M. Ruess, D. Shillinger, E. Rank. *FCMLab: A Finite Cell Research Toolbox for MATLAB*. Advances in Engineering Software, 2014.

[8] C. Davila, P. Camanho and A. Turon. *Effective Simulation of Delamination in Aeronautical Structures Using Shells and Cohesive Elements*. Journal of Aircraft.

[9] P. Rahulkumar, A. Jagota, S.J. Bennison and S. Saigal. *Cohesive element modeling of viscoelastic fracture: application to peel testing of polymers*. International Journal of Solids and Structures, 2000.

[10] D. Schillinger and M. Ruess. *The Finite Cell Method: A Review in the Context of Higher-Order Structural Analysis of CAD and Image-Based Geometric Models*. Arch Computational Methods Eng, 2015.